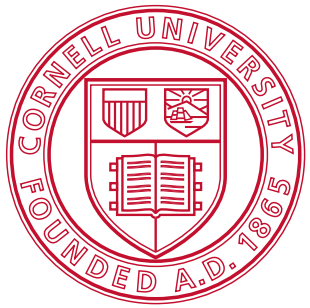
Lecture 11

Image Processing

Erdal Yılmaz



Cornell University

July 29, 2013

Before we begin

HW3 solutions

HW4 will be assigned today

PR Projects

Today is Lena Day

- Colors, RGB
- Image files
- MATLAB Demos

Pixel and Bitmaps

- Pixel = Picture Element
- Bitmaps store color information of pixels

Representation of Colors

- Black and White
- Grayscale
- Red Green Blue
- Number of Colors?

Anatomy of Files

- Text Files vs. Binary Files
- Example Text files:

  .txt (plain), .html (markup)
- Example Binary files:

  .exe .bmp .png .zip

X-raying Binary Files

- Use hexdump
- File formats
- File headers

X-raying BMP Files

- Size in pixels?
- Number of colors?

MATLAB Image Formats

imformats

shows information about the image formats MATLAB can work on

imfinfo

shows information related to an image file

Reading Image Files

imread

reads an image file and returns image data in an array

Usage

```
% A = imread(FILENAME, FORMAT)

data = imread('lena.png','png');
```

Displaying Image Data

imshow

displays the image in a figure window. if RGB layers are provided it is displayed in colors. If a single layer is used, it will be displayed in grayscale.

Usage

```
imshow('lena.png');

[x,map] = imshow('lena.tif');
imshow(x,map);

I = imread('lena.png');
imshow(I);
```

[Red, Green, Blue] Lena

```
l = imread('lena.png');

r = l;
r(:,:,2:3) = 0;
figure
imshow(r)

g = l;
g(:,:,1:2:3) = 0;
figure
imshow(g)

b = l;
b(:,:,1:2) = 0;
figure
imshow(b)
```

[Red, Green, Blue] Lena

Converting to Gray Scale

togray.m

```
data = imread('lena.png');

size(data) % 3D array! ( 512 x 512 x 3 )

r = data(:,:,1);
g = data(:,:,2);
b = data(:,:,3);

gray = (r/3+g/3+b/3);

data(:,:,1) = gray;
data(:,:,2) = gray;
data(:,:,3) = gray;

imshow(data)
```

Converting to Gray Scale

Writing Image Files

imwrite

writes the image to a file

Usage

```
% imwrite(A, FILENAME, FORMAT)

imwrite(data, 'lena_gray.png', 'png');
```

Filtering

We will apply a 3x3 matrix which is called a filter to process the image.

Averaging and Blurring

$$F_a = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} /9 \qquad F_b = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} /5$$

Applying the Filter

```
function pdata = filter_lena(fname, fmt, F)
data = imread([fname,'.',fmt], fmt);
[n m c] = size(data);
pdata = zeros(n,m,c);
ddata = double(data); %convert from uint8 to double
% ignoring a thin frame around the picture
for i = 2:n-1    % move to the next row
 for j = 2:m-1  % move to the next column
  for k = 1:3  % apply the filter to all 3 layers
   pdata(i,j,k) = ...
          sum(sum(F .* ddata(i-1:i+1, j-1:j+1, k)));
  end
 end
end
pdata = uint8(floor(pdata)); %convert back to uint8
imwrite(pdata, [fname,'_filtered.',fmt], fmt);
```

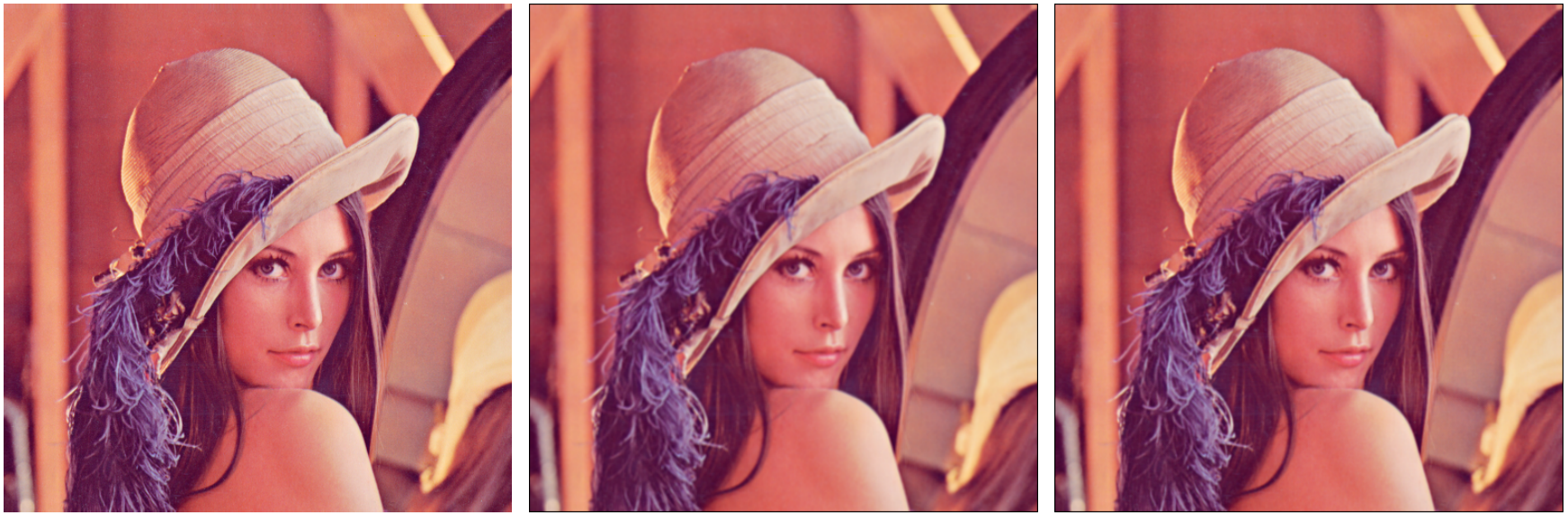
Applying the Filter

```
Fa = ones(3)/9;
Fb = ones(3)/5; Fb([1 3],[1 3]) = 0;

I  = imread('lena.png');
Ia = filter_lena('lena','png',Fa);
Ib = filter_lena('lena','png',Fb);

subplot(1,3,1); imshow(I)
subplot(1,3,2); imshow(Ia)
subplot(1,3,3); imshow(Ib)
```

Blurred Images



Edge Detection

$$F_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \qquad F_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

$$D_x = F_x .* I \qquad D_y = F_y .* I \qquad |\nabla I| = \sqrt{D_x^2 + D_y^2}$$

Edge Detection